

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-137605

(43)Date of publication of application : 16.05.2000

(51)Int.Cl.	G06F 9/06
	H04M 3/42
	H04Q 3/545

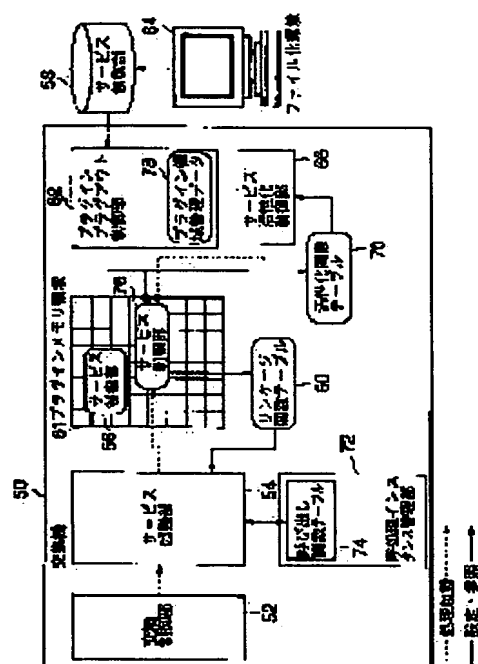
(21)Application number : 10-312181 (71)Applicant : FUJITSU LTD
(22)Date of filing : 02.11.1998 (72)Inventor : ISHITA FUMIYASU
YAMADA HIROTOSHI
KAMURO SHINGO

(54) EXCHANGE SERVICE CONTROL METHOD, EXCHANGE AND RECORDING MEDIUM RECORDING EXCHANGE SERVICE CONTROL PROGRAM

(57)Abstract:

PROBLEM TO BE SOLVED: To add/change/delete exchange service without stopping an exchange during an operation by referring to a pointer held by a second table, executing a function which the pointer shows and supplying service.

SOLUTION: A service control part is loaded and the entry address of activation program is recorded in an activation table 70. A service activation control part 66 refers to the entry address recorded in an activation function table 70 and starts a service control part 76. The service control part 76 records the pointer of a function executing service in a linkage function table 60. The pointer of the function obtained by referring to the linkage function table 60 is recorded in the call function table 74 of a call processing instance management part 72 by a service starting part 54. The service starting part 54 refers to the function pointer recorded in the call function table 74, executes the function in the service control part and executes service.



LEGAL STATUS

[Date of request for examination] 22.12.2000

[Date of sending the examiner's decision of rejection] 01.11.2005

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

【特許請求の範囲】

【請求項1】プログラムにより制御される交換機において、

特定のサービスを実行するサービスプログラムがローディングされると、

所定の場合に該サービスプログラムは該サービスプログラムが有する該サービスを実行するための関数へのポインタを第1のテーブルに記録し、

該交換機は、

発呼により該サービスの利用要求があると呼毎に一時的な記憶領域を確保し、

該第1のテーブルを参照して得た該ポインタを該記憶領域内の第2のテーブルに保持し、

該第2のテーブルに保持された該ポインタを参照することにより該ポインタの示す関数を実行してサービスを提供することを特徴とする交換機サービス制御方法。

【請求項2】前記交換機は、

前記サービスプログラムがローディングされる記憶領域の領域空塞状態及び前記サービスプログラムの状態を管理するデータを有し、

該データを用いて前記ローディングを制御する請求項1記載の交換機サービス制御方法。

【請求項3】前記所定の場合は、

前記交換機が周期的に前記ローディングを監視して前記サービスプログラムがローディングされたことを検出した場合、または、コマンドにより指示が入力された場合である請求項1記載の交換機サービス制御方法。

【請求項4】前記サービスプログラムは独立してコンパイルされて前記ローディングされる請求項1記載の交換機サービス制御方法。

【請求項5】既存のサービスプログラムを別のサービスプログラムに置き換える際に該既存のサービスプログラムを実行中の第1の呼が有る場合、

該別のサービスプログラムによるサービスを提供すべき第2の呼が発生すると、

前記交換機は新たに前記一時的な記憶領域を確保して該別のサービスプログラムの有する関数へのポインタのテーブルを該記憶領域に保持し、

該ポインタの示す関数を実行することにより該第2の呼に該別のサービスプログラムによるサービスを提供する請求項1記載の交換機サービス制御方法。

【請求項6】ローディングされた前記サービスプログラムを前記交換機から削除する際、

前記第1のテーブルの値を該サービスプログラムがローディングされる前の値に戻す請求項1記載の交換機サービス制御方法。

【請求項7】プログラムにより制御される交換機であって、

特定のサービスを実行するサービスプログラムをローディングする手段と、

該サービスプログラムが有する該サービスを実行するための関数へのポインタを保持する第1のテーブルを有し、

所定の場合に該サービスプログラムに該関数へのポインタを該第1のテーブルに記録させる手段と、

発呼により該サービスの利用要求がある場合に呼毎に一時的な記憶領域を確保する手段と、

該第1のテーブルを参照して得た該ポインタを該記憶領域内の第2のテーブルに保持する手段と、

該第2のテーブルに保持された該ポインタを参照することにより該ポインタの示す関数を実行してサービスを提供する手段とを有することを特徴とする交換機。

【請求項8】前記サービスプログラムがローディングされる記憶領域の領域空塞状態及び前記サービスプログラムの状態を管理するデータを有し、

該データを用いて前記ローディングを制御する手段を有する請求項7記載の交換機。

【請求項9】前記交換機は周期的に前記ローディングを監視する手段とコマンド入力手段を有し、

前記所定の場合は、

該監視する手段により前記サービスプログラムがローディングされたことを検出した場合、または、コマンドにより指示が入力された場合である請求項7記載の交換機。

【請求項10】既存のサービスプログラムを別のサービスプログラムに置き換える際に既存のサービスプログラムを実行中の第1の呼が有る場合、

該別のサービスプログラムによるサービスを提供すべき第2の呼が発生した場合に前記交換機は新たに前記一時的な記憶領域を確保して該別のサービスプログラムの有する関数へのポインタのテーブルを該記憶領域に保持する手段と、

該ポインタの示す関数を実行することにより該第2の呼に該別のサービスプログラムによるサービスを提供する手段とを有する請求項7記載の交換機。

【請求項11】ローディングされた前記サービスプログラムを前記交換機から削除する場合に前記第1のテーブルの値を該サービスプログラムがローディングされる前の値に戻す手段を有する請求項7記載の交換機。

【請求項12】プログラムにより制御される交換機のサービスを制御するコンピュータ読み取り可能な交換機サービス制御プログラムを記録した記録媒体であって、該交換機に特定のサービスを実行するサービスプログラムをローディングし、

所定の場合に該サービスプログラムに該サービスプログラムが有する該サービスを実行するための関数へのポインタを第1のテーブルに記録させ、

発呼により該サービスの利用要求があると呼毎に一時的な記憶領域を確保し、

該第1のテーブルを参照して得た該ポインタを該記憶領

域内の第2のテーブルに保持し、
該第2のテーブルに保持された該ポインタを参照することにより該ポインタの示す関数を実行してサービスを提供することを特徴とする交換機サービス制御プログラムを記録した記録媒体。

【請求項13】前記サービスプログラムがローディングされる記憶領域の領域空塞状態及び前記サービスプログラムの状態を管理するデータを保持し、
該データを用いて前記ローディングを制御する請求項12記載の交換機サービス制御プログラムを記録した記録媒体。

【請求項14】前記所定の場合は、
前記交換機が周期的に前記ローディングを監視して前記サービスプログラムがローディングされたことを検出した場合、または、コマンドにより指示が入力された場合である請求項12記載の交換機サービス制御プログラムを記録した記録媒体。

【請求項15】既存のサービスプログラムを別のサービスプログラムに置き換える際に既存のサービスプログラムを実行中の第1の呼が有る場合、
該別のサービスプログラムによるサービスを提供すべき第2の呼が発生すると、
前記交換機は新たに前記一時的な記憶領域を確保して該別のサービスプログラムの有する関数へのポインタのテーブルを該記憶領域に保持し、
該ポインタの示す関数を実行することにより該第2の呼に該別のサービスプログラムによるサービスを提供する請求項12記載の交換機サービス制御プログラムを記録した記録媒体。

【請求項16】ローディングされた前記サービスプログラムを前記交換機から削除する際、
前記第1のテーブルの値を該サービスプログラムがローディングされる前の値に戻す請求項12記載の交換機サービス制御プログラムを記録した記録媒体。

【請求項17】プログラムにより制御される交換機のサービスを制御するコンピュータ読み取り可能な交換機サービス制御プログラムを記録した記録媒体であって、
該交換機に特定のサービスを実行させる関数と、
該交換機にローディングされた場合に該関数へのポインタを該交換機内のテーブルに記録するプログラムを含み、
該交換機により該テーブルに記録された該ポインタが参照されて該関数が実行されることを特徴とする交換機サービス制御プログラムを記録した記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、交換機にサービス機能を追加等する技術に係り、特に、交換機が提供中のサービスを中断させることなくサービス機能を追加・変更・削除する交換機サービス制御方法及び交換機及び交

換機サービス制御プログラムを記録した記録媒体に関する。

【0002】近年、通信市場開放に伴い通信事業者間の競争が激化しており、通信事業者は加入者への提供サービスにより差別化を図っている。しかし、サービスが多様化するにつれ、そのサービスを実現する交換機におけるソフトウェアの複雑さが一層増したため、短期間でかつ容易に新しいサービスを提供できない現状がある。したがって、特に通信事業者において、新規サービスを導入する場合であっても運転を停止することはもとより、既存サービスへの影響を与えずに新規サービスの機能を追加できる交換機が求められている。

【0003】

【従来の技術】図1は、従来技術による交換機の交換ソフトウェア実行モジュール10を示すモジュール図である。同図に示すように、交換ソフトウェア実行モジュール10はプログラム基盤となるオペレーティングシステム20、共通機能プログラムモジュール群16、保守運用プログラムモジュール群18、交換制御プログラムモジュール群14、サービス制御プログラムモジュール群12から構成される。共通機能プログラムモジュール群16はメモリのエリア獲得等の共通機能を有する共通機能プログラム1～xから構成され、保守運用プログラムモジュール群18は保守運用機能を有する保守運用プログラム1～nから構成され、交換制御プログラムモジュール群14は呼接続等の交換機の基本機能を有する交換制御プログラム1～mから構成され、サービス制御プログラムモジュール群12は3者通話等のサービス提供機能を有するサービス制御プログラム1～kから構成される。また、サービス制御プログラムと交換制御プログラムは互いにリンケージを取りながら処理を実行する。

【0004】従来の技術によると、オペレーティングシステムや保守運用機能を含めた、全ての交換サービスを制御する機能モジュールを上述した一つの交換ソフトウェア実行モジュールとして交換機にローディングして運用している。従って、運用中の交換サービスに新しい機能を付加する場合、特に付加の規模が大きい場合や全く新規のサービス制御プログラムを追加しなければならない場合には、交換機システム全体のソフトウェアを稼働中の交換機システムを一時停止させて入れ替える方法がとられていた。また、付加する機能の規模が大きい場合には、図2に示すように稼働中の交換機30におけるサービス制御部34あるいは交換制御部32のプログラムの一部にパッチと呼ばれる部分変更を加えることで新しい機能を実現してきた。なお、サービス制御部34では前述のサービス制御プログラムが実行され、交換制御部32では前述の交換制御プログラムが実行される。図2中、パッチ生成環境46にてソフトウェアパッチ44が生成され、前述の保守運用プログラムの一部が実行されるパッチ管理部42を介してパッチ領域36に追加

サービスプログラム 38、40 が書き込まれ、交換制御部 32 及びサービス制御部 34 には、追加サービスプログラムに実行を移し、実行が終われば戻る旨のパッチ 33、35 が書き込まれる。

【0005】

【発明が解決しようとする課題】しかしながら、上記の従来の方法によると、サービス追加の際に提供中のサービスや通話に影響を与えてしまうという問題点があった。すなわち、パッチによる機能追加の場合は、稼働中の交換機にパッチが投入された直後から追加されたプログラムが動作するため、呼処理データが変更される場合等には、パッチを投入する前に呼び出し中や通話中になっている呼がパッチ投入後に切断される可能性があった。また、パッチによると呼処理のデータを大幅に変更追加することが困難であるために、追加できる機能に制限がある。更に、パッチでは対応できない機能追加の場合には、新規プログラムを交換機全体のプログラムと共にコンパイルし直し、交換機を停止させて交換機のソフトウェア全体の入れ替えをする必要があった。

【0006】本発明は上記の点に鑑みなされたもので、稼働中の交換機を停止させることなく、更にサービス実施中における呼び出し中・通話中等の呼状態に影響を与えずに、交換サービスを追加・変更・削除する交換機サービス制御方法及び交換機及び交換機サービス制御プログラムを記録した記録媒体を提供することを目的とする。

【0007】

【課題を解決するための手段】上記課題を解決するための本発明の構成は、以下の通りである。請求項 1 に記載の発明は、プログラムにより制御される交換機において、特定のサービスを実行するサービスプログラムがローディングされると、所定の場合に該サービスプログラムは該サービスプログラムが有する該サービスを実行するための関数へのポインタを第 1 のテーブルに記録し、該交換機は、発呼により該サービスの利用要求があると呼毎に一時的な記憶領域を確保し、該第 1 のテーブルを参照して得た該ポインタを該記憶領域内の第 2 のテーブルに保持し、該第 2 のテーブルに保持された該ポインタを参照することにより該ポインタの示す関数を実行してサービスを提供することを特徴とする交換機サービス制御方法である。

【0008】本発明により、動的に呼出関数へのリンクージュをとることが可能となるので、コンパイル時での呼び出し先と呼び出し元とのアドレスリンクージュが不要となり、サービス追加等の場合に従来のようにパッチあるいは全体ファイル作成・入れ替えが不要となるため、サービス追加が容易となりサービスの中断も防止できる。また、サービス運用中にサービスプログラムを改造したサービスプログラムを再びローディングしても、サービス実施中の全く独立したサービスプログラムの呼は

もとより、改造前のサービスプログラムの呼にも全く影響を与えずにサービスプログラムをローディングできる。

【0009】請求項 2 に記載の交換機サービス制御方法は、前記交換機が、前記サービスプログラムがローディングされる記憶領域の領域空塞状態及び前記サービスプログラムの状態を管理するデータを有し、該データを用いて前記ローディングを制御する。本発明により、ローディングされるサービスプログラムの記憶領域が既存のサービスプログラムと衝突することを防止することができる。また、サービスプログラムを削除する場合にサービス実施中のサービスプログラムを不用意に削除できないよう制御することが可能になる。

【0010】請求項 3 に記載の交換機サービス制御方法は、前記所定の場合が、前記交換機が周期的に前記ローディングを監視して前記サービスプログラムがローディングされたことを検出した場合、または、コマンドにより指示が入力された場合である。本発明により、ローディングしたプログラムを迅速に使用することができる。

【0011】請求項 4 に記載の交換機サービス制御方法は、前記サービスプログラムが独立してコンパイルされて前記ローディングされることである。本発明により、サービスプログラム開発が容易になる。請求項 5 に記載の交換機サービス制御方法は、既存のサービスプログラムを別のサービスプログラムに置き換える際に該既存のサービスプログラムを実行中の第 1 の呼が有る場合、該別のサービスプログラムによるサービスを提供すべき第 2 の呼が発生すると、前記交換機は新たに前記一時的な記憶領域を確保して該別のサービスプログラムの有する関数へのポインタのテーブルを該記憶領域に保持し、該ポインタの示す関数を実行することにより該第 2 の呼に該別のサービスプログラムによるサービスを提供する。

【0012】本発明により、改造サービスの提供等の場合において、従来のように改造前のサービスを使用中の呼が切断されることなく、サービスを入れ替えることができる。請求項 6 に記載の交換機サービス制御方法は、ローディングされた前記サービスプログラムを前記交換機から削除する際、前記第 1 のテーブルの値を該サービスプログラムがローディングされる前の値に戻す請求項 1 に記載の交換機サービス制御方法。

【0013】本発明により、ローディングしたサービスプログラムを削除しても、呼の切断を発生させずにローディング前の状態でサービスを継続することが可能となる。以下の発明も上記と同様の効果を有する。請求項 7 に記載の発明は、プログラムにより制御される交換機であって、特定のサービスを実行するサービスプログラムをローディングする手段と、該サービスプログラムが有する該サービスを実行するための関数へのポインタを保持する第 1 のテーブルを有し、所定の場合に該サービスプログラムに該関数へのポインタを該第 1 のテーブルに

記録させる手段と、発呼により該サービスの利用要求がある場合に呼毎に一時的な記憶領域を確保する手段と、該第 1 のテーブルを参照して得た該ポインタを該記憶領域内の第 2 のテーブルに保持する手段と、該第 2 のテーブルに保持された該ポインタを参照することにより該ポインタの示す関数を実行してサービスを提供する手段とを有することを特徴とする交換機である。

【0014】請求項 8 に記載の交換機は、前記サービスプログラムがローディングされる記憶領域の領域空塞状態及び前記サービスプログラムの状態を管理するデータを有し、該データを用いて前記ローディングを制御する手段を有する。請求項 9 に記載の交換機は、前記交換機が周期的に前記ローディングを監視する手段とコマンド入力手段を有し、前記所定の場合が、該監視する手段により前記サービスプログラムがローディングされたことを検出した場合、または、コマンドにより指示が入力された場合である。

【0015】請求項 10 に記載の交換機は、既存のサービスプログラムを別のサービスプログラムに置き換える際に既存のサービスプログラムを実行中の第 1 の呼が有る場合、該別のサービスプログラムによるサービスを提供すべき第 2 の呼が発生した場合に前記交換機は新たに前記一時的な記憶領域を確保して該別のサービスプログラムの有する関数へのポインタのテーブルを該記憶領域に保持する手段と、該ポインタの示す関数を実行することにより該第 2 の呼に該別のサービスプログラムによるサービスを提供する手段とを有する。

【0016】請求項 11 に記載の交換機は、ローディングされた前記サービスプログラムを前記交換機から削除する場合に前記第 1 のテーブルの値を該サービスプログラムがローディングされる前の値に戻す手段を有する。請求項 12 に記載の発明は、プログラムにより制御される交換機のサービスを制御するコンピュータ読み取り可能な交換機サービス制御プログラムを記録した記録媒体であって、該交換機に特定のサービスを実行するサービスプログラムをローディングし、所定の場合に該サービスプログラムに該サービスプログラムが有する該サービスを実行するための関数へのポインタを第 1 のテーブルに記録させ、発呼により該サービスの利用要求があると呼毎に一時的な記憶領域を確保し、該第 1 のテーブルを参照して得た該ポインタを該記憶領域内の第 2 のテーブルに保持し、該第 2 のテーブルに保持された該ポインタを参照することにより該ポインタの示す関数を実行してサービスを提供することを特徴とする交換機サービス制御プログラムを記録した記録媒体である。

【0017】請求項 13 に記載の交換機サービス制御プログラムを記録した記録媒体は、前記サービスプログラムがローディングされる記憶領域の領域空塞状態及び前記サービスプログラムの状態を管理するデータを保持し、該データを用いて前記ローディングを制御する。請

求項 14 に記載の交換機サービス制御プログラムを記録した記録媒体は、前記所定の場合が、前記交換機が周期的に前記ローディングを監視して前記サービスプログラムがローディングされたことを検出した場合、または、コマンドにより指示が入力された場合である。

【0018】請求項 15 に記載の交換機サービス制御プログラムを記録した記録媒体は、既存のサービスプログラムを別のサービスプログラムに置き換える際に既存のサービスプログラムを実行中の第 1 の呼が有る場合、該別のサービスプログラムによるサービスを提供すべき第 2 の呼が発生すると、前記交換機は新たに前記一時的な記憶領域を確保して該別のサービスプログラムの有する関数へのポインタのテーブルを該記憶領域に保持し、該ポインタの示す関数を実行することにより該第 2 の呼に該別のサービスプログラムによるサービスを提供する。

【0019】請求項 16 に記載の交換機サービス制御プログラムを記録した記録媒体は、ローディングされた前記サービスプログラムを前記交換機から削除する際、前記第 1 のテーブルの値を該サービスプログラムがローディングされる前の値に戻す。請求項 17 に記載の発明は、プログラムにより制御される交換機のサービスを制御するコンピュータ読み取り可能な交換機サービス制御プログラムを記録した記録媒体であって、該交換機に特定のサービスを実行させる関数と、該交換機にローディングされた場合に該関数へのポインタを該交換機内のテーブルに記録するプログラムを含み、該交換機により該テーブルに記録された該ポインタが参照されて該関数が実行されることを特徴とする交換機サービス制御プログラムを記録した記録媒体である。

【0020】

【発明の実施の形態】図 3 は本発明の実施例である交換機 50 における機能構成の概要を示す機能ブロック図である。同図に示すとおり、交換機 50 は、交換制御を行う交換制御部 52、サービス制御部を起動するサービス起動部 54、サービスを実行するサービスプログラムであるサービス制御部 56 及び 76、サービス制御部 58 をシステム内に組み込むためのプラグインプラグアウト制御部 62、サービス制御部がローディングされる領域であるプラグインメモリ領域 61、プラグインメモリ領域 61 におけるメモリの空塞状態を管理するプラグイン領域管理データ 78、サービス制御部を活性化させるためのサービス活性化制御部 66 および活性化関数テーブル 70 を有する。また、既存のサービス制御部を、改良したサービス制御部に置き換える場合や、サービス制御部を削除すなわちプラグアウトする場合にサービス実施中の呼に影響を与えないための機能であるリンケージ関数テーブル 60、呼処理インスタンス管理部 72、呼び出し関数テーブル 74 を有する。また、新たなサービス制御部 58 を生成するために、ファイル化環境 64 が用いられる。

【0021】次に、図4に示すフローチャートを用いて本実施例の動作の概要を説明する。ステップ1としてファイル化環境64にてサービスの入れ替え、追加のためのサービス制御部58を生成する。なお、サービス制御部は実行できる形式のプログラムであり、光ディスクや磁気テープ等の媒体に記録されている。ステップ2として、サービス制御部58はプラグインプラグアウト制御部62に読み込まれ、ステップ3として読み込まれたサービス制御部はプラグインメモリ領域61の空きの領域に取り込まれる。ここで、プラグインメモリ領域61のメモリの空塞状態はプラグイン領域管理データ78により管理されており、プラグイン領域管理データ78をプラグインプラグアウト制御部62が参照することにより、サービス制御部がプラグインメモリ領域61に取り込まれる。

【0022】サービス制御部がローディングされると同時にステップ4として活性化関数テーブル70に活性化用プログラムのエントリーアドレスが記録される。すなわち、サービス制御プログラムに組み込まれ、サービス制御部を活性化させるための関数（処理プログラム）のエントリーアドレスを活性化関数テーブル70に記録する。なお、活性化とはサービス制御部がサービス起動部54により起動がかけられる状態にすることをいう。

【0023】ステップ5として、サービス活性化制御部66が活性化関数テーブル70に記録されたエントリーアドレスを参照し、サービス制御部76に起動をかける。ステップ6として起動をかけられたサービス制御部76は、サービスを実行する関数のポインタをリンク関数テーブル60に記録する。ステップ7としてサービス起動部54がリンク関数テーブル60を参照する。ステップ8としてサービス起動部54がリンク関数テーブル60を参照して得られた関数のポインタを呼処理インスタンス管理部72の呼び出し関数テーブル74に記録する。そして、ステップ9としてサービス起動部54は呼び出し関数テーブル74に記録された関数ポインタを参照してサービス制御部における関数を実行してサービスを実行する。なお、呼び出し関数テーブル74はダイヤルによりサービスがスタートした時点に呼処理インスタンス管理部72に生成されるメモリ領域に設けられる。

【0024】また、呼び出し関数テーブル74はサービスの切り替え、追加時等に使用される。すなわち、改良サービスに切り替えるような場合であって、改良前のサービス制御部が発呼を受けて動作中であるとき、サービス起動部はリンク関数テーブルから改良サービスの関数のアドレスもらい、それを別の呼び出し関数テーブルに記録し、その後そのサービスの発呼を受けた場合は、サービス起動部は新たな呼び出し関数テーブルを参照して改良サービスを実行する。サービスの切り替え、追加時等の動作の詳細については後述する。

【0025】図5は本発明の実施例における交換ソフトウェア実行モジュールを示すモジュール図である。同図に示すように、交換ソフトウェア実行モジュール80はプログラム基盤となるオペレーティングシステム82、共通機能プログラムモジュール群84、保守運用プログラムモジュール群86、交換制御プログラムモジュール群88、サービス起動プログラムモジュール群90、リンク関数テーブル92、実行モジュール94₀～94_nから構成される。なお、実行モジュール94₀はオペレーティングシステム82、共通機能プログラムモジュール群84、サービス起動プログラムモジュール群90、リンク関数テーブル92をまとめた部分である。

【0026】交換制御プログラムモジュール群88、オペレーティングシステム82、共通機能プログラムモジュール群84、保守運用プログラムモジュール群86、交換制御プログラムモジュール群88は従来の技術と同様である。サービス起動プログラムモジュール群90は、サービス起動プログラム1～nにより構成される。サービス起動プログラムは交換制御プログラムから起動がかかった後に、クロスリンク関数テーブル92を参照しながら、サービス制御プログラムを決定し、起動する。実行モジュール94₁～94_nはサービス制御プログラムと関数テーブル生成プログラムから構成され、サービス制御プログラムは、実行するプログラムへの入りの関数を関数表に備えている。また、活性化した場合に、関数テーブル生成プログラムは、リンク関数テーブルに入り口の関数のポインタを設定する。このようなソフトウェアモジュール構成をとることで、交換制御プログラム群を含む実行モジュール94₀とサービス制御部の実行モジュール94₁～94_nを独立させ、サービスを新規に作成する場合や既存サービスを改造する場合など、既存部分である実行モジュール94₀に影響を与えずに独立してコンパイルし実行モジュールを生成できる。従って、ファイル化環境64において他のサービスとは独立したサービス制御部を得ることができる。

【0027】図6は本発明の実施例である交換機のハードウェア構成図である。同図に示すように交換機100は全体の制御を行う主制御装置(MPR: Main Processor)110、呼の制御を行う呼制御装置(CPR: Call Processor)120、130、140から構成される。主制御装置110は、プログラムの処理等を行う中央処理装置(CPU)111、中央処理装置(CPU)111に実行させるためのプログラム及びデータを記憶する主記憶装置112、交換機外部とのインターフェースをとるための入出力装置113、プログラム及びデータを記憶する外部記憶媒体114、光磁気ディスク等の可搬記憶媒体と情報をやりとりするための外部記憶装置115、各制御装置間で情報をアクセスする際の制御を行うプロセッサ間アクセス

制御装置 116 から構成される。呼制御装置 120 は外部記憶媒体 121、プロセッサ間アクセス制御装置 122、中央処理装置 123、主記憶媒体 124、通話路装置 125 から構成される。他の呼制御装置も同様の構成である。上記で説明した、本発明の実施例における機能は主に主制御装置 110 内の中央処理装置 111 において実行される。

【0028】図 7 は、本発明の実施例である交換機 50 の機能構成の詳細を示す機能ブロック図である。本実施例の主要部は前述した概要構成と同様であるので各部の詳細を説明する。サービス起動部 54 は n 個のサービス起動プログラム 55₁ ~ 55_n を内蔵している。呼処理インスタンス管理部 72 は、起呼時に呼処理に必要なメモリブロックを捕捉する。このメモリブロック内には、呼毎に呼び出し関数テーブル 74 が割りつけられる。サービス制御部 56 にはサービス制御プログラム 57 と関数テーブル生成プログラム 67 が内蔵されており、サービス制御部 76 にはサービス制御プログラム 77 と関数テーブル生成プログラム 79 が内蔵されている。プラグインプラグアウト制御部 62 はプラグインメモリ領域管理データ 78、サービス制御部をローディングする Plug-in Loader 69 及びサービス制御部がローディングされるプラグインメモリ領域（図 7 には記載を略している）内の領域をプラグイン領域管理データを参照して決定する Plug-in Manager 68 を有している。サービス活性化制御部 66 は活性化プログラム 65 を有し、サービス活性化要求があった場合に活性化関数テーブル 70 を参照する。

【0029】以下、図 7 及びフローチャート等を用いて本発明の実施例の動作について詳細に説明する。図 8 は、図 7 に示す実施例において、サービスプラグインと称しているサービス制御部のローディングについての処理を示すフローチャートである。ファイル化環境 64 で生成されたサービス制御部 58 は、ステップ 1 として外部記憶装置 115 から読み出され（図 7（a））、ステップ 2 として一旦外部記憶媒体 114 に保存される。ステップ 3 として Plug-in Loader 69 は Plug-in Manager 68 にローディング先のアドレスを問い合わせる。ステップ 4 として Plug-in Manager 68 は、サービス制御部がリロケータブル実行形式すなわちどのメモリアドレスでも動作可能な実行形式であるかどうかを調べ、リロケータブルであればステップ 5 としてプラグイン領域管理データ 78 を基にして空き領域を補足する。ステップ 4 においてサービス制御部がリロケータブル実行形式でない場合、例えば絶対番地実行形式すなわちコンパイル時点で実行するメモリアドレスが決定されている形式の場合には、ステップ 6 としてプラグイン領域管理データ 78 を基にして Plug-in Manager 68 によってコンパイラが割り付けた実行アドレスがプラグインメモリ領域の空エリアであるかどうかを検証される。ステップ 7

として、検証結果が二重割り付けかどうか判断され、二重割り付けであればステップ 8 として警告メッセージが出力されプラグインが拒否される。ステップ 7 において二重割り付けでなければステップ 9 に進む。ステップ 5 とステップ 8 の後もステップ 9 に進む。ステップ 9 においてサービス制御部のメモリ割付が完了したかどうかチェックされ、完了していない場合は、サービスプラグイン処理を終了する。完了している場合は、ステップ 10 として、サービス制御部をロードするメモリ領域アドレスを決定し、ステップ 11 として、割り付け状態を表す割り付け状態テーブルを“プラグイン済”の状態を設定する。

【0030】ステップ 3 においてアドレス問い合わせが終了すると、ステップ 12 として、メモリ領域が二重割り付けかどうかチェックし、二重割り付けならば処理を終了する。二重割り付けでなければ、ステップ 13 として、捕捉されたメモリ領域アドレスにサービス制御部をロードし（図 7（b））、ステップ 14 としてロードしたサービス制御部の関数テーブル生成プログラムのアドレスが活性化関数テーブル 70 に書き込まれ（図 7（c））処理を終了する。

【0031】図 9 はプラグイン領域管理データ 78 の例であり、プラグインメモリ領域の空塞管理はビットマップ形式の（a）に示すようなメモリ空塞テーブルで管理される。このテーブルでの 1 ビットは 8KB のエリアを意味し、“1”が使用中、“0”が空きを示す。サービス制御部がリロケータブル実行形式の場合、プラグインするサービス制御部のサイズを 8KB で割り、格納可能なエリアサイズを求め、連続して空いているエリアすなわち連続して“0”が存在するビット位置を求めることにより格納アドレスを決定する。サービス制御部が絶対番地実行形式である場合には、コンパイラが割り付けたサービス制御部の先頭番地からメモリ空塞テーブルのビット位置をもとめ、そのサイズ分が空きエリアであるかをチェックする。

【0032】図 9（b）はプラグイン領域管理データの割り付け状態テーブルを示し、システムに実装可能な最大サービス分の情報が管理できるようにあらかじめエリアが確保されている。このテーブルは主にプラグインメモリ領域の状態表示機能に使用される。例えば、プラグインメモリ領域にどのようなサービスが実装されているかやプラグインメモリ領域の使用状況を表示するコマンド等で利用されている。このため、割り付け状態テーブルにはサービス名称やプラグインメモリ領域位置やその専有サイズ、さらにそのサービスの状態を保持している。また、この割り付け状態テーブルに前述のリロケータブル形式か絶対番地形式かのプログラム属性の情報を保持することで、プラグインメモリ領域の空きエリアが点にしまった場合等、例えばプラグイン／プラグアウトを繰り返すことによりメモリ空塞テーブル上空きビット

が点在してしまった場合等に、リロケータブル形式のサービス制御部を再割り付けし、空きエリアを拡大することができる。

【0033】以上の処理によって、サービス制御部58はプラグインプラグアウト制御部62でプラグインメモリ領域上にローディングされる。図10に活性化関数テーブル70の例を示す。活性化関数テーブル70には、サービス制御部の識別番号であるサービス番号によって得られるサービス制御部内の関数テーブル生成プログラムへのアドレスが設定される。このテーブルは後述するサービス活性化制御部66で参照される。

【0034】図11は、サービス制御部の活性化とリンク関数テーブル生成の処理を示すフローチャートである。ステップ1としてサービス制御部がプラグインメモリ領域にローディングすなわちプラグインされると、ステップ2としてサービス活性化制御部66は活性化関数テーブル70より、ロードされたサービス制御部内の関数テーブル生成プログラムのアドレスを得て、ステップ3として関数テーブル生成プログラムを起動(図7(d))して関数テーブル生成処理を開始し、ステップ4としてリンク関数テーブルを読み出す。ステップ5としてリンク関数テーブル60に設定されたサービスが新規サービスかどうかをチェックし、新規サービスならばステップ7に進み、新規サービスでなければ、ステップ6として旧リンク関数テーブルの内容を保存してからステップ7に進む。ステップ7において、サービス制御プログラムの全関数をリンク関数テーブルに設定する(図7(e))。ステップ8として設定完了かどうかをチェックし、完了していなければステップ9として設定異常警告を発して処理を終了し、設定が完了していればそのまま処理を終了する。

【0035】ステップ3により関数テーブル生成プログラムを起動すると、ステップ10として関数テーブルの生成が完了したかどうかをチェックし、完了していなければ処理を終了し、完了していればステップ11としてプラグイン領域管理データの割り付け状態テーブルにおける割付状態を”実装サービス中”に変更して処理を終了する。

【0036】サービス活性化、リンク関数テーブル生成について図7を用いて更に説明する。図7のサービス制御部56、76は、他の実行モジュールと独立してコンパイルされているため、プラグインメモリ領域上にローディングされても他の実行モジュールからは呼び出すことができない。このため、サービス活性化制御部66の活性化プログラム65によりサービス制御部76を活性化すなわち使用可能状態にするのである。

【0037】活性化プログラム65は周期起動またはコマンドにより起動される。周期起動の場合には、定期的にプラグインプラグアウト制御部62内のプラグインメモリ領域管理データ78を監視することで、プラグイン

されたサービス制御部76のサービス番号を知ることができる。また、コマンドで起動する場合には直接保守者からサービス番号を入力するか、または入力されたサービス名称からプラグインメモリ領域管理データ78を利用してサービス番号を得ることができる。このようにして得たサービス制御部76のサービス番号と活性化関数テーブル70から、サービス制御部76内の関数テーブル生成プログラム79に起動をかける(図7の(d))。活性化プログラム65から起動された関数テーブル生成プログラム79は、サービス制御プログラム77内に定義されている関数表をもとにリンク関数テーブル60に関数のアドレスを設定するのである(図7の(e))。

【0038】図12は、リンク関数テーブル60の構成と関数テーブル生成プログラム79の動作を示した図である。図12において、リンク関数テーブル60はサービス番号でインデックスされる初段テーブルと関数番号でインデックスされる関数ポインタテーブルで構成されている。初段テーブルには、関数ポインタテーブルのアドレスが書かれており、関数ポインタテーブルには、サービス制御プログラム77の関数ポインタアドレスが設定できるようになっている。サービス制御部76内の関数テーブル生成プログラム79は、コンパイル時に生成されているサービス制御プログラム77の関数表およびサービス番号をもとにリンク関数テーブル60の関数ポインタテーブルに関数ポインタを設定する。このように設定されたリンク関数テーブル60はサービス起動プログラムによって参照される。

【0039】図7において、あるサービスが起動された場合に交換制御部52はそのサービスに対応する特定のサービス起動プログラム、ここではサービス起動プログラム55₂を呼び出す(図7の(f))。サービス起動プログラム55₂はまず呼処理インスタンス管理部72からメモリブロックを捕捉する。その後、前述のように設定されたリンク関数テーブル60から関数情報を読み出し(図7の(h))、捕捉されたそのメモリブロック内の呼出関数テーブル74に関数アドレスを保存する(図7の(g))。サービス起動プログラム55₂からサービス制御プログラム77を起動する場合(図7の(i))には、この呼出関数テーブル74が使用される。

【0040】すなわち、交換制御部52から起動されるサービス起動プログラムは、直接サービス制御プログラムの関数を呼び出すのではなく、サービス番号に従ってリンク関数テーブルから取得した関数ポインタテーブル150(図12)を、一旦呼処理インスタンス管理部72内の呼制御用メモリブロックに呼び出し関数テーブル74として保存し、その保存した呼び出し関数テーブルを参照してサービス制御プログラムの関数160(関数1~3)を呼び出す。

【0041】図13は上記の処理をフローチャートで示した図である。同図において、あるサービスの起呼時、ステップ1として交換制御部からサービス起動プログラムに起動がかかり、ステップ2としてサービス起動プログラムは呼処理インスタンス管理部72のメモリエリアを捕捉する。ステップ3としてリンケージ関数テーブルから関数のアドレスを読み出し、ステップ4としてその読み出したアドレスを、呼処理インスタンス管理部72のメモリエリア内の呼制御メモリブロックにコピーすることにより呼び出し関数テーブル74を設定する。ステップ5としてサービス起動プログラムは呼び出し関数テーブル74を読み出す。また、サービスにより、ダイヤル数字受信、接続応答、切断等に応じて、呼処理インスタンスメモリエリア内の読み出し関数テーブルを読み出す。ステップ6として、読み出した関数アドレスが有効かどうかをチェックし、有効でなければステップ7として呼の強制開放等の異常時処理を行い処理を終了し、有効であればステップ8としてサービス制御プログラムの起動を行い処理を終了する。ステップ8においては、ダイヤル数字受信、接続応答、切断等の受信信号の種類に応じて呼び出し関数テーブルから決定されたサービス制御部のサービス制御プログラムが呼び出される。

【0042】上述の通り本発明によれば、

(1) 全く新しいサービスとしてのサービス制御部を追加する場合であっても、リンケージ関数テーブル72に新しいサービス制御プログラムの関数を設定することで機能追加ができる。

(2) 既存サービスを改造する場合であっても、改造したサービス制御部を新しいサービス同様に追加してリンケージ関数テーブルを書き換えることにより新旧を入れ換えることができる。この時、古いサービス制御部で動作している呼があっても発呼時にサービス起動プログラムが補足した呼制御用メモリブロック内にサービス制御部の関数ポインタが保存されているため、通話中の呼に影響を及ぼすことなく処理を実行できる。

【0043】以上の一連の処理の流れを、フリーダイヤルサービスがある加入者から発呼された場合を例にとり図14～16を用いて説明する。まずステップ1としてシステム初期設定を行う。すなわち図14に示すように、サービス制御部内の関数テーブル生成プログラムがシステムの初期設定時にサービス制御プログラムに実装されたフリーダイヤルサービスを実行する関数(f1とf2)のポインタアドレスをリンケージ関数テーブルに設定する。

【0044】図15において、フリーダイヤルサービスがある加入者から発呼された時、ステップ2として、交換制御プログラムからサービス起動プログラムに起動がかけられ、ステップ3としてサービス起動プログラムが呼処理インスタンス管理部内で呼制御メモリブロックを補足し、ステップ4としてリンケージ関数テーブルの内

容を呼制御メモリブロックにコピーする。呼制御メモリブロックは、その呼が終了するまでの間その内容を保持できるメモリ領域である。ステップ5として呼制御メモリブロックに保持された関数テーブルを参照し、ステップ6としてサービス起動プログラムからサービス制御プログラムが実行される。

【0045】ここで、更に機能追加したフリーダイヤルサービスが追加される場合を図16に示す。サービス制御プログラムとしてローディングされた新たなフリーダイヤルサービスをフリーダイヤル2と称し、関数としてはf1'とf2'を有することとする。フリーダイヤル2に対応するサービス制御プログラムをローディングした時に関数テーブル生成プログラムが起動され、ステップ7として、追加された関数f1'とf2'のエントリアドレスがリンケージ関数テーブルに上書き設定される。呼処理インスタンス管理部内の呼制御メモリブロック上に前のサービス制御プログラムへの関数(f1とf2)が保持され、サービス制御プログラムがステップ8として参照しているため、既にサービスを開始している加入者は、新サービスプログラムがローディングされても影響を受けない。

【0046】再び新しい加入者からの発呼を交換制御プログラムが受け付け、ステップ9としてサービス起動プログラムが起動された場合には、ステップ10としてサービス起動プログラムによって新しい呼制御メモリブロックが補足され、ステップ11としてf1'とf2'を含む関数テーブルが呼制御メモリブロックにコピーされ、ステップ12としてサービス起動プログラムが関数へのアドレスを参照して、ステップ12として呼処理サービスが実行される。この場合、新旧のサービス制御プログラムが同時に動作しているが、やがて旧サービスプログラムを使用している加入者が終話することで古いサービス制御プログラムが使用されなくなる。

【0047】以上、サービスの追加、切り替え、すなわちサービスプラグインについて説明した。次に、一度プラグインしたサービス制御部をプラグイン前の状態に戻すサービスプラグアウトについて説明する。プラグインしたサービス制御部を単にプラグアウトするだけでは実行プログラムが削除されサービスが停止するばかりか、システムに重大な影響を与えることになる。そこで、プラグアウト時には、下記の手順に従う。

【0048】(1) プラグアウトすべきサービス制御部への発呼を停止する。

(2) 既に通話中すなわちサービス実施中の呼を監視し、すべての呼が終話した時点でサービス制御プログラムをプラグアウトする。

具体的なプラグアウト処理の手順を図17のフローチャートおよび図7の機能ブロック図を用いて説明する。

【0049】プラグアウト時には、ステップ1としてサービス活性化制御部66内にサービスの閉塞をコマンド

で通知する。ステップ2として活性化プログラム65は閉塞通知をPlug-in Manager 68に渡す(図7

(j))。ステップ3としてPlug-inManager 68はプラグイン領域管理データ78内の割り付け状態テーブルの状態値を”プラグアウト要求中”に書き換える。ステップ4として、Plug-in Manager 68は定期的にこの割り付け状態テーブルを検索し、プラグアウト要求中のサービスについて現在通話中の呼が存在するかを呼処理インスタンス管理部72に問い合わせる(図7(k))。ステップ5として通話中呼がないかを判定し、通話中呼があればステップ6としてしばらく待ってからステップ5に戻り、通話中呼が存在しない場合には、ステップ7としてプラグイン領域管理データ78内の割り付け状態テーブルの該当サービスの状態値を初期値に戻し、ステップ8として閉塞が完了したかどうかをチェックし、完了していなければ処理を終了し、完了していればステップ9として、呼び出し関数テーブルのリカバリとしてステップ10からの処理を行う。

【0050】ステップ10として活性化プログラム65は関数テーブル生成プログラム79を起動する。ステップ11として関数テーブル生成プログラム79は旧リンクエージ関数テーブルの読み出しを行い、ステップ12としてテーブルがあるかどうかをチェックし、テーブルがなければステップ13としてリンクエージ関数テーブルを初期設定し、テーブルがあればステップ14として旧リンクエージ関数テーブルの内容でリンクエージ関数テーブルを設定して、関数テーブルのリカバリの処理を終了する。すなわち、関数テーブル生成プログラムは該当するリンクエージ関数テーブル60の関数ポインタを書き換え前の状態(新規サービスであれば”0”に、既存サービス入れ換えであれば既存サービスの関数ポインタ値)に設定するのである。したがって、交換制御部52からサービス起動プログラムが起動された時に、関数ポインタ値が”0”であればサービス制御プログラムは起動されず、新しく発呼される呼は規制される。また、既存サービスへのリンクエージ関数テーブルの値が関数ポインタである場合には、そのポインタを使用して既存サービスが起動されるため、サービスの中断が発生しない。

【0051】次に、ステップ15として活性化関数テーブル70のリカバリを行い、ステップ16としてサービス制御部をプラグインメモリ領域上から取り除く。交換機サービス制御プログラムを記録した記録媒体は図6に示す主記憶装置112又は外部記憶媒体114のいずれか一方または両方に相当するものである。上記の記録媒体には上述した説明の手順に従い中央処理装置111を動作させるためのプログラムが格納されており、既存の交換機のメモリ部等にこのプログラムを格納することにより、既存の交換機を本発明の交換機として使用することができる。主記憶装置112又は外部記憶媒体114を実現する記録媒体としては、電子メモリやハードディ

スク等がある。また、本発明の交換機サービス制御プログラムを格納した光磁気ディスク、磁気テープ等から既存の交換機のメモリ部等にこのプログラムをローディングすることにより既存の交換機を本発明の交換機として使用することができる。

【0052】なお、本発明は、上記の実施例に限定されことなく、特許請求の範囲内で種々変更・応用が可能である。

【0053】

10 【発明の効果】本発明により、交換機の動作中でのサービスの入れ替えが可能になる。従来はサービスを入れ替えるためにシステム全体のファイル生成及びシステムスタート時の再開(初期化)動作が必要となり、サービスの一時中断が発生していたが、本発明により、交換機全体の動作に影響を与えることなく、該当するサービスの切り替えおよび追加を可能になる。また、今後の交換機の技術動向の一つとなり得る「ユーザ・プログラマビリティ」の実現のために本発明は有効である。すなわち、これまで交換機ユーザ(電話会社等)は加入者付加サービスなどの新機能追加作業を交換機ベンダーに依存して行っていたが、本発明のようにサービスに影響を与えることなく機能追加できることとなれば、ユーザ自身が新機能の開発を行ってサービス機能追加をより短期間で行うことが可能となる。本発明はその基盤技術となり得るものである。

【図面の簡単な説明】

【図1】従来技術による交換ソフトウェア実行モジュールを示すモジュール図である。

20 【図2】従来技術においてパッチを用いてサービス機能を追加する例である。

30 【図3】本発明の実施例における交換機の機能構成概要を示す機能ブロック図である。

【図4】図3に示す構成の動作を示すフローチャートである。

【図5】本発明の実施例における交換ソフトウェア実行モジュールを示すモジュール図である。

【図6】本発明の実施例における交換機の構成を示すブロック図である。

40 【図7】本発明の実施例における交換機の機能構成詳細を示す機能ブロック図である。

【図8】図7に示す実施例において、サービス制御部のローディングについての処理を示すフローチャートである。

【図9】プラグイン領域管理データ78の例である。

【図10】活性化関数テーブル70の例である。

【図11】サービス制御部の活性化とリンクエージ関数テーブル生成の処理を示すフローチャートである。

【図12】リンクエージ関数テーブル60の構成と関数テーブル生成プログラム79の動作を示した図である。

50 【図13】あるサービスの起呼が発生した場合の交換制

御部とサービス起動プログラムによる処理を示したフローチャートである。

【図 14】初期設定を示す図である。

【図 15】フリーダイヤルサービスがある加入者から発呼された場合を示す図である。

【図 16】新たに機能追加したフリーダイヤルサービスが追加される場合を示す図である。

【図 17】プラグアウト処理の手順を示すフローチャートである。

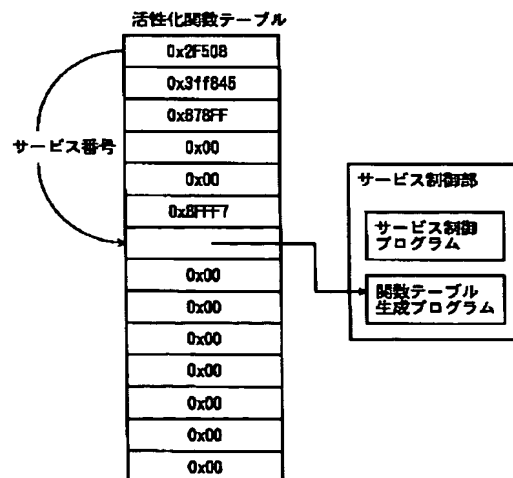
【符号の説明】

10、80 交換機ソフトウェア実行モジュール
 12 サービス制御プログラムモジュール群
 14、88 交換制御プログラムモジュール群
 16、84 共通機能プログラムモジュール群
 18、86 保守運用プログラムモジュール群
 20、82 オペレーティングシステム
 30 交換機
 32 交換制御部
 33、35 パッチ
 34、56、58、76 サービス制御部
 36 パッチ領域
 38、40 追加サービスプログラム
 42 パッチ管理部
 44 ソフトウェアパッチ
 46 パッチ生成環境
 50、100 交換機
 52 交換制御部
 54 サービス起動部
 55 サービス起動プログラム

57、77 サービス制御プログラム
 60、92 リンケージ関数テーブル
 61 プラグインメモリ領域
 62 プラグインプラグアウト制御部
 64 ファイル化環境
 65 活性化プログラム
 66 サービス活性化制御部
 67 関数テーブル生成プログラム
 68 Plug-in Manager
 69 Plug-in Loader
 70 活性化関数テーブル
 72 呼処理インスタンス管理部
 74 呼び出し関数テーブル
 77 サービス制御プログラム
 78 プラグイン領域管理データ
 79 関数テーブル生成プログラム
 90 サービス起動プログラムモジュール群
 94 実行モジュール
 110 主制御装置
 111、123 中央処理装置
 112、124 主記憶装置
 113 入力装置
 114、121 外部記憶媒体
 115 外部記憶装置
 116、122 プロセッサ間アクセス制御装置
 120、130、140 呼制御装置
 125 通話路装置
 150、160 関数 1～3

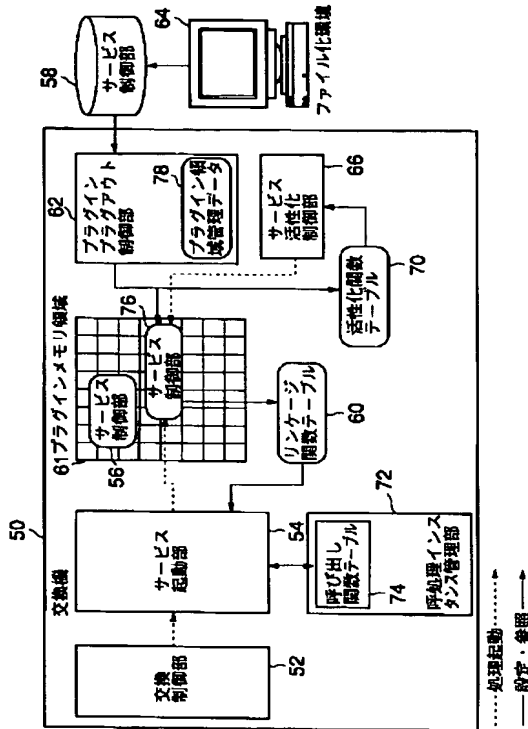
【図 10】

活性化関数テーブル 70 の例



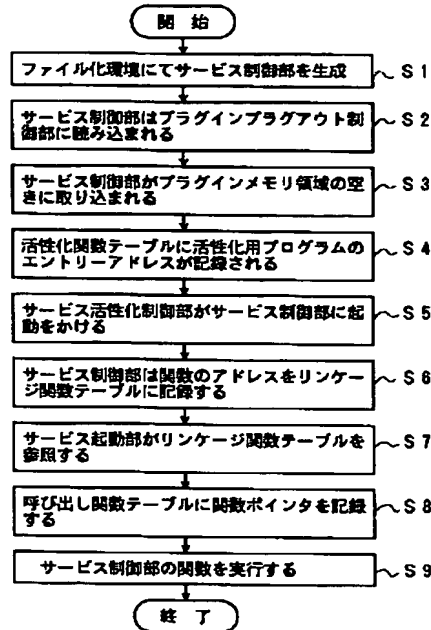
【図 3】

本発明の実施例における交換機の機能構成概要を示す機能ブロック図



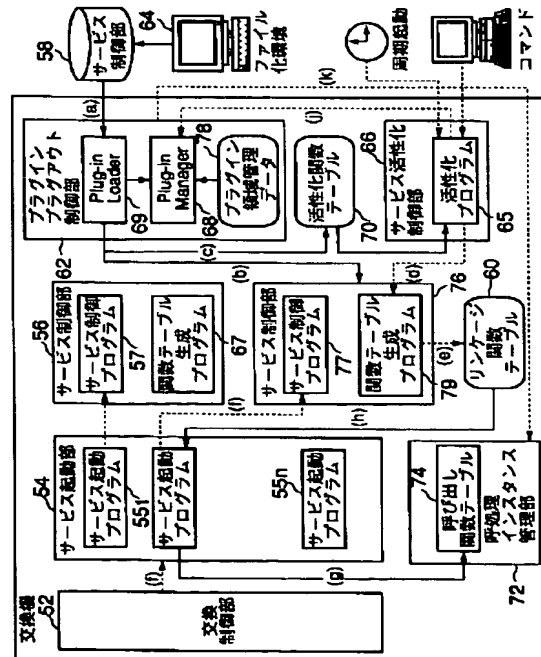
【図 4】

図 9 に示す構成の動作を示すフローチャート



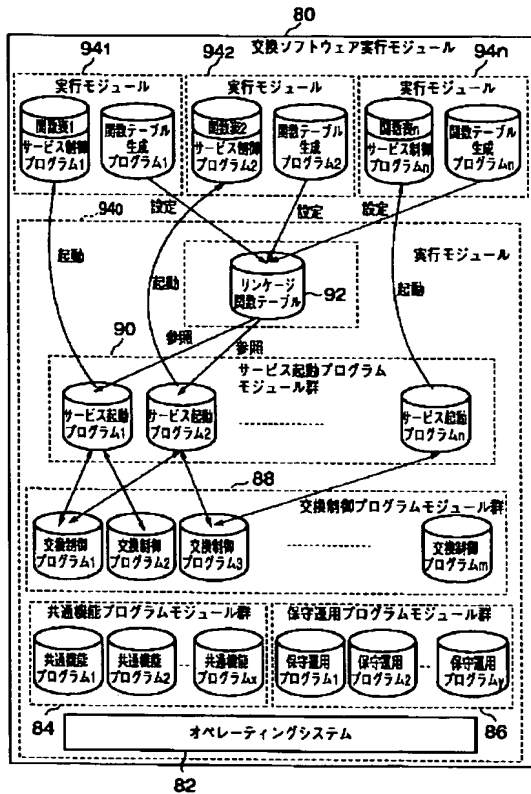
【図 7】

本発明の実施例における交換機の機能構成詳細を示す機能ブロック図



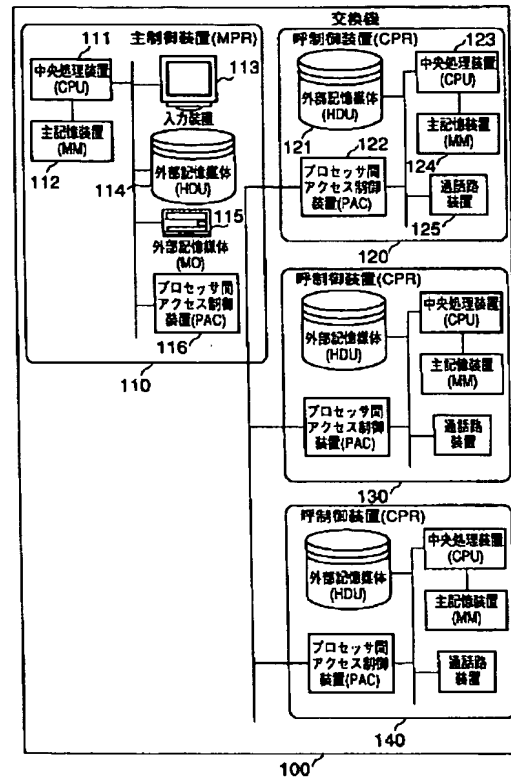
【図 5】

本発明の実施例における交換ソフトウェア
実行モジュールを示すモジュール図



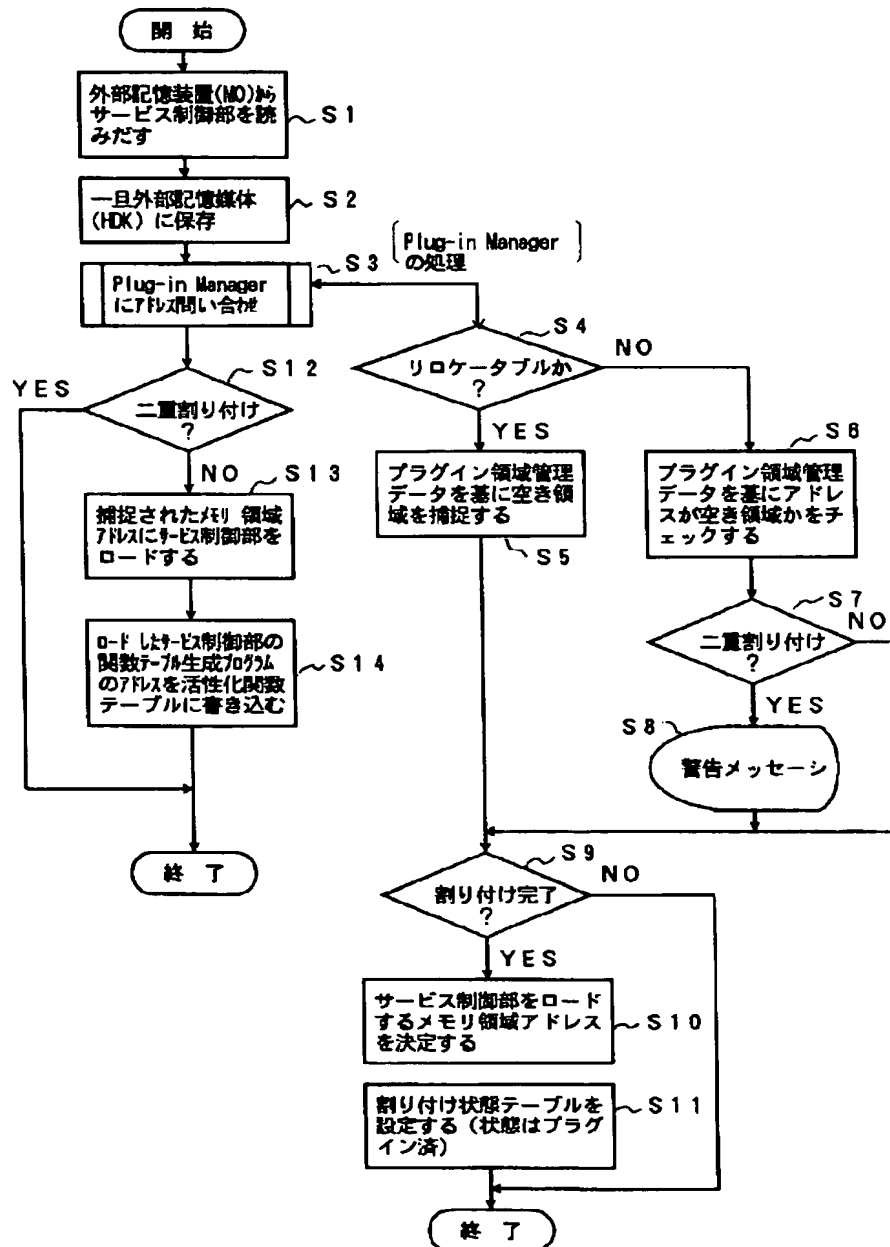
【図 6】

本発明の実施例における交換機の構成を示すブロック図



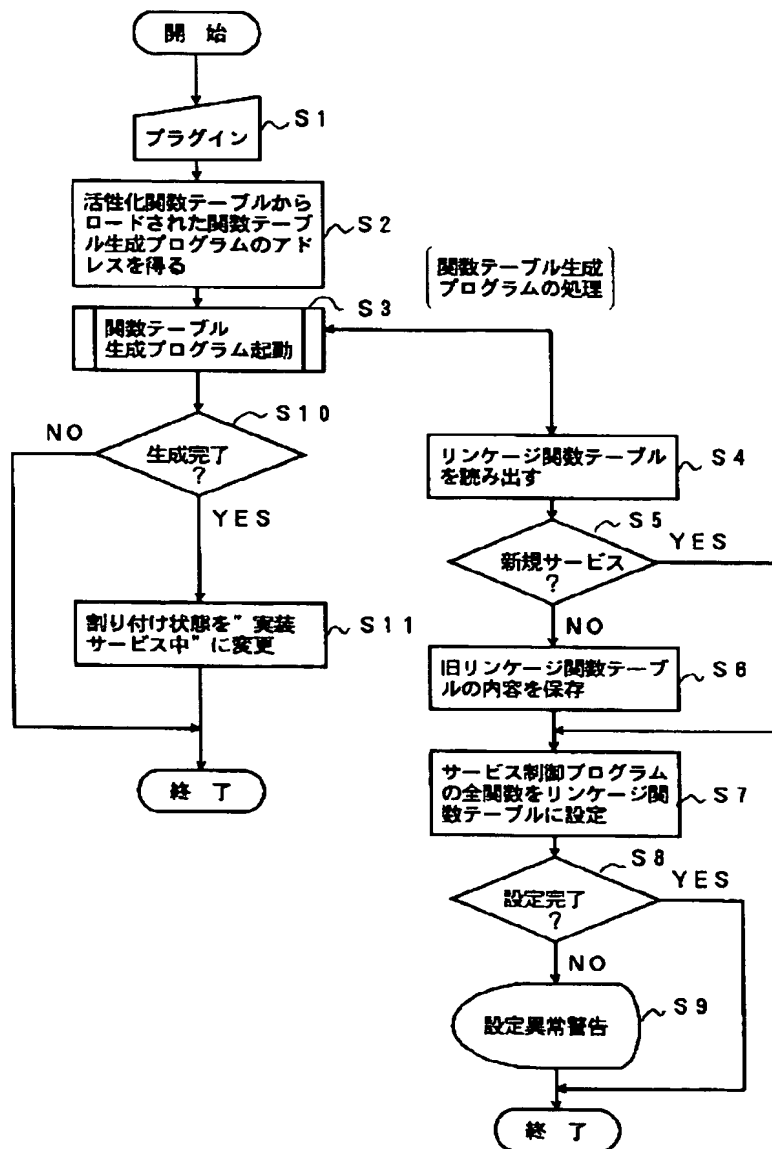
【図 8】

図 7 に示す実施例において、サービス制御部のローディング
 についての処理を示すフローチャート



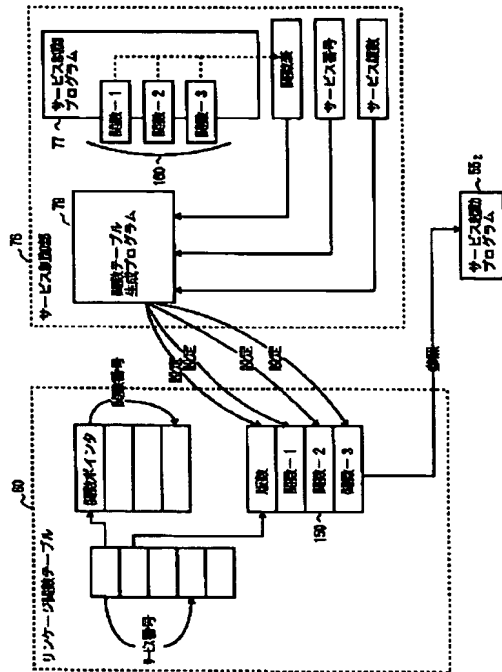
【図 11】

サービス制御部の活性化とリンケージ関数
テーブル生成の処理を示すフローチャート



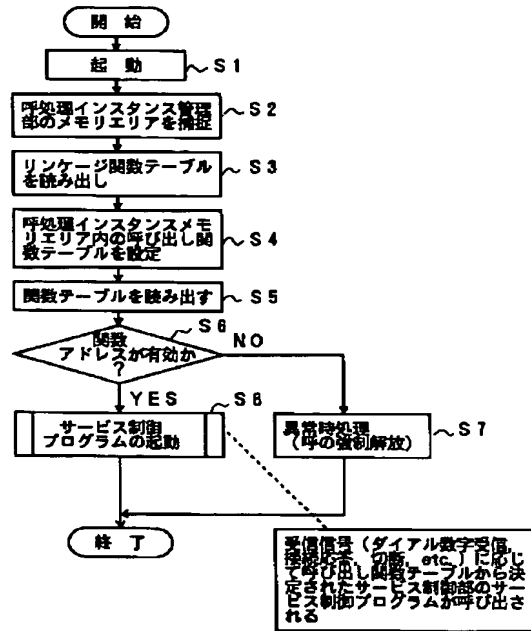
【図 12】

リンケージ関数テーブル 80 の構成と関数テーブル
生成プログラム 79 の動作を示した図



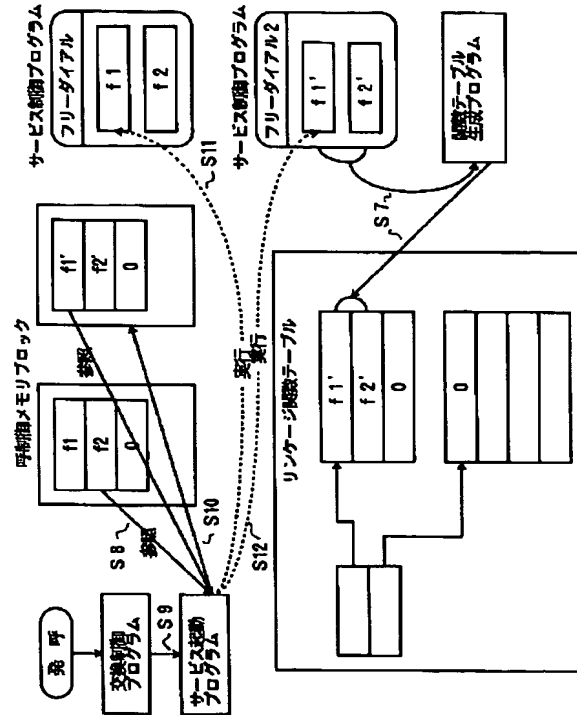
【図 13】

あるサービス起呼が発生した場合の交換制御部とサービス
起動プログラムによる処理を示したフローチャート



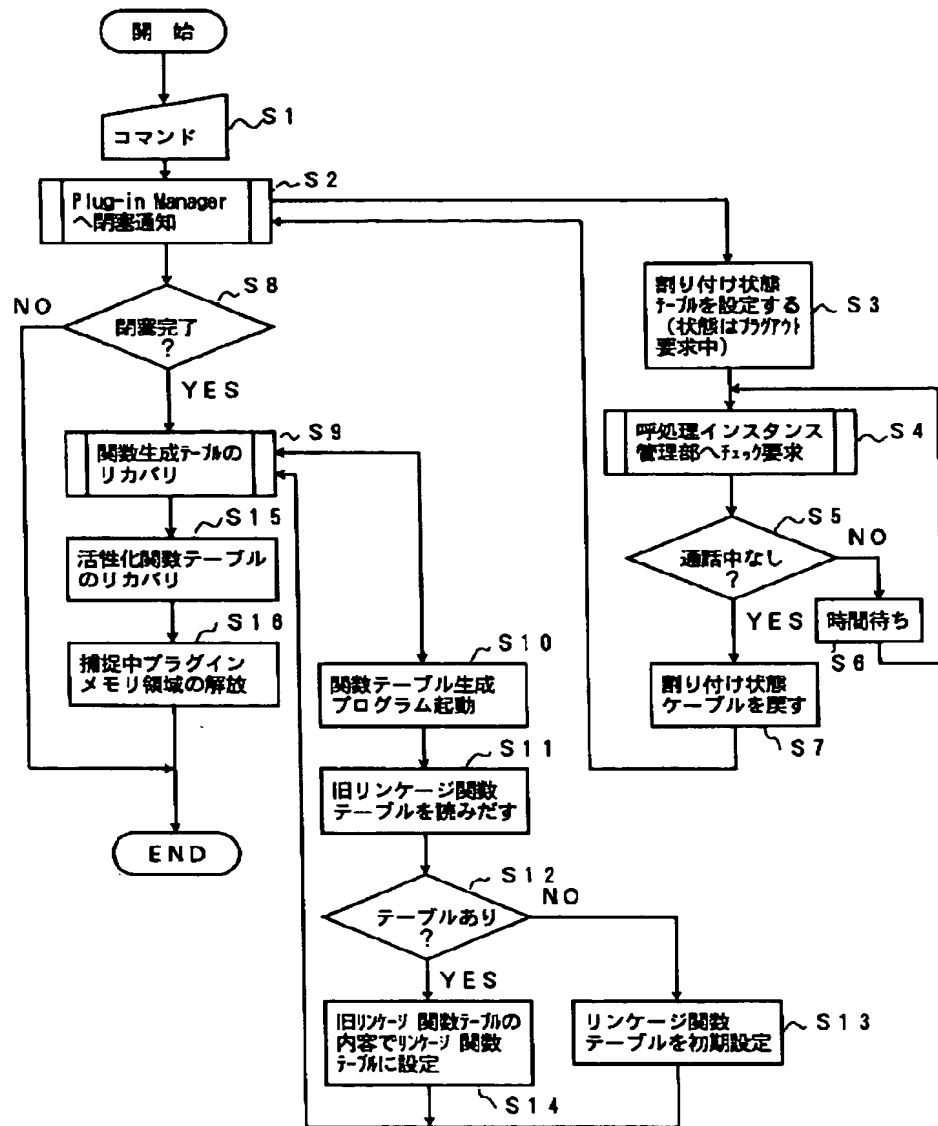
【図16】

新たに機能追加したフリーダイヤルサービスが追加される場合を示す図



【図 17】

プラグアウト処理の手順を示すフローチャート



フロントページの続き

(72)発明者 加室 真吾
 神奈川県川崎市中原区上小田中4丁目1番
 1号 富士通株式会社内

Fターム(参考) 5B076 EA17
 5K024 AA00 AA45 BB04 GG00
 5K026 AA10 BB00 CC07 FF02 FF03
 FF04 FF23 GG12 GG15
 9A001 CC02 CC07 JJ12 LL05 LL09